# Design of Internet Connectivity for Mobile Ad hoc Networks

Erik Nordström
Department of Information Technology
Uppsala University
erik.nordstrom@it.uu.se

Per Gunningberg
Department of Information Technology
Uppsala University
per.gunningberg@it.uu.se

Christian Tschudin
Computer Science Department
University of Basel
christian.tschudin@unibas.ch

## Abstract

We present a design space analysis of the problem of providing Internet connectivity for mobile ad hoc networks (MANETs). Currently there are a plethora of proposals to solve this problem in the research community. However, we argue that many of the existing designs suffer from complexity and design solutions that have not been properly evaluated. One reason for this is the lack of implementations. We believe that a design space analysis will help to clear the field and lead to better designs. We illustrate this approach by presenting a new system design for MANET Internet connectivity. We have evaluated our system in simulation and have found that not only is it more simple than common existing approaches, but also more efficient and less likely to fail in the face of routing updates. In addition, we have implemented our system in a real testbed to illustrate its ability to provide a complete mobility solution together with Mobile IP.

## 1 Introduction

In recent years, routing protocol implementations for mobile ad hoc networks (MANETs) have become increasingly abundant, but practical experiences from real world scenarios are still limited. One explanation is that standard scenarios tend to be too limited in scope or that people are not convinced about their applicability to reality. Internet connectivity is a potential service that could increase the benefit of MANETs and also make the application scenarios more relevant. However, one obstacle on the way to reach that goal is the lack of consensus in the research community on what it concretely means to have Internet connectivity in a MANET. It is clear that it means that ad hoc nodes should be able to establish communication with hosts in the Internet, but does it also mean that those nodes should, for example, be able to move within or between networks, change between multiple gateways or use several at once, or that nodes should be globally reachable or not? Therefore, the problem we consider in this paper is *the design of Internet connectivity for MANETs that can handle node mobility, both within and inbetween networks, having continuous and uninterrupted Internet connections whenever there is at least one potential route to one or more gateways*. Without this clear view of the problem of Internet connectivity it is difficult or impossible to design a solution that is robust, efficient and that in the end solves the problem because the problem is not clearly specified.

Herein lies also the problem with the state of MANET Internet connectivity in general as we see it. When we have tried to implement Internet connectivity for MANETs we found that many of the proposals out there are confusing and difficult to understand and evaluate. That is often because they are not clear about what problem they try to solve, other than the loosely defined problem of Internet connectivity. Another reason is complexity. Proposals, typically in the form of Internet drafts, try to cover all angles by stating they support multiple solutions to the same problem, e.g., forwarding strategy. The interactions between different system elements and respective choice of solution are therefore difficult to predict. The drafts have generally not been implemented or evaluated, other than sometimes in simulation. Hence the soundness of their approach is not proven and the assumptions and design choices lead to fragile designs or designs that are not applicable in reality. One concrete example of this is the commonly suggested usage of a default route in a MANET that under some circumstances, as we show later, makes nodes experience stalled TCP connections in the face of multiple gateways to the Internet.

The question then is, how can we avoid designs that suffer from problems like this and in the end construct more robust Internet connectivity designs for MANETs? In this paper we present one approach that we believe answers this question to satisfaction. This approach consists of, through a problem diagnosis, defining a design

space that aims to provide reference points for the analysis and evaluation of existing design proposals, as well as to improve the quality of new designs. Our hope is that this design space will give researchers in the field a more coherent view of the problems to solve, the trade-offs in design choices and in the end lead to less effort spent on divergent proposals.

The primary contribution of this paper is our diagnosis and presentation of the design space for Internet connectivity. A second contribution is the description, implementation and evaluation of a complete system for Internet connectivity based on this diagnosis. Our design is robust in that it works in very challenging scenarios and hence less stringent ones too, whilst achieving an acceptable trade-off between performance and overhead. In addition to robustness, our design is simple and flexible by supporting, e.g, multi-homing and interfacing to Mobile IP while still requiring minimal modifications to existing routing protocols. The design uses tunneling to achieve indirection (non shortest path routing) and has been integrated with an implementation of the AODV [15] routing protocol. The design is evaluated in simulation by comparing it to another common proposal using a standard default route also implemented by us. Our results show that our design is more robust, achieves better performance and is more flexible. Since our implementation also works in the real systems, we provide results from real world experiments to illustrate the interfacing with Mobile IP.

The rest of the paper is structured as follows. In section 2 we introduce and diagnose the general principles and problems of providing Internet connectivity for MANETs that provides the input for defining our design space. The following section 3 reviews related work in the context of this diagnosis. In section 4 we define the design space for MANET Internet connectivity. The following section 5 describes our system for Internet connectivity in MANETs that we have designed based on our design space analysis. Section 6 reports on results from evaluating our system in simulation through a comparison to another competing design. We also provide results from real world experiments showing our design's applicability to reality. Section 7 concludes the paper with a discussion.

# 2 Problem Diagnosis

In this section we diagnose the problem of Internet connectivity in MANETs to provide the necessary input in order to define our design space. Our diagnosis is based on an as challenging scenario as possible, without being unrealistic. That is because the design space is defined by all possible design solutions and a more challenging scenario means more solutions. We start by decomposing the problem of Internet connectivity in ad hoc networks into three sub problems:

> i) *Determining a node's location*, ii) *Discovering gateways* and iii) *Establishing and maintaining consistent forwarding states to gateways.*

The natures of these problems are different depending on the assumptions for the specific scenario. Unless the scenario is very specific or there is an administrative entity in the network, it is hard to make any assumptions on what the network looks like. An ad hoc network is, by definition, to some degree unmanaged. Under those circumstances it is not possible to assume that there is, for example, only one gateway, that nodes move in a certain way or that nodes use a specific prefix for their configured IP address. Hence, we argue that a general Internet connectivity solution must be robust enough to handle the most most challenging scenarios. We define such a scenario with the following assumptions:

1. There might be multiple gateways to the Internet
2. Nodes are mobile, at both micro and macro scales
3. The routing protocol is reactive and hop-by-hop, i.e., each node has a limited horizon in the view of the network and only knows the next hop towards a destination.
4. Nodes do not share a common IP-prefix

To list a few example applications where we believe this type of scenario makes sense we point to RoboCupRescue [3], a competition to build autonomous relief robots where some teams use ad hoc networking to communicate between the robots. The robots could relay telemetry information onto the Internet or another fixed network where it can be processed by rescue crews. Here robots are deployed quickly in a environment that is unknown and hostile, hence it is not possible to assume anything less than the worst case. Other potential applications are remote surveillance [4] or planetary exploration where a set of autonomous mobile robots collaboratively explore the surface of a distant planet, relaying telemetry information to one or more orbiting satellites part of the inter-planetary Internet [10]. Here it is necessary to assume the worst case, because there is no way to easily change the system after deployment. Another motivation for a challenging scenario is that a design for such a scenario also works in less stringent ones, but the same might not be true for a system designed for less stringent scenarios in the first place and hence would be limited in its applicability. We now motivate each of the assumptions 1-4 and describe their implications on the sub problems i), ii) and iii) above.

**Multiple Gateways.** Since every node is a potential router and there is no sole administrator, a node might also be a gateway. Any node with an Internet connection could potentially offer that service to other nodes in the ad hoc network if it so wishes. Multiple gateways have implications for problem ii) in that discovering several gateways gives the option to either select one gateway at a time or use several at once. For iii) in that a TCP connection might break if the forwarding state is suddenly re-pointed to another gateway somewhere along a path without the explicit knowledge of the source of the connection.

**Mobility.** For the second point we argue that nodes might be (micro) mobile within a MANET, but they should also be able to seamlessly move between different MANETs and be (macro) mobile between a MANET and the Internet. The latter assumption might require, e.g., Mobile IP [14] and hence integration with the Internet connectivity system. The mobility assumption also has implications on i) and iii). Agent registration must match that of the currently used gateway and if a route switches to another gateway, the source nodes using that route must be notified so that they can re-register with the new agent there.

**Routing.** The mobility assumption implies a routing protocol that reacts swiftly to topology changes. The implications of reactiveness for problem i) are that the protocol only maintains a partial network state (routes to active destinations only). Therefore, in combination with prefix-less addressing, there is no way to easily determine node locations, i.e., whether a node is located in the MANET or in the Internet. For ii) it is important that the Internet connectivity design supports reactive gateway discovery. The partial network view of the routing protocol in combination with hop-by-hop forwarding is a problem for iii). Each hop on the forwarding path runs the risk of repeating the problem of determining node locations for every packet.

**Addressing.** Prefix-less or *flat* addresses is a common assumption in the ad hoc network research community and is a requirement for macro mobility. A node should, in line with the Mobile IP specification, be able to bring its preconfigured home address into the ad hoc network and use it for routing. Hence, there is no common prefix among nodes and the ad hoc network is flat in both a routing and addressing sense. As mentioned above, this has implications for problem i) in combination with reactive routing. Using a proactive protocol or prefixes/subnets solve the problem since node locations can be determined either by checking the routing table or by examining the IP address prefix of the destination address.

In addition to the functionality for operation in the worst case scenario, an Internet connectivity design could offer optional functionality for flexibility, for example, exploiting multiple gateways for the purpose of multi-homing or load balancing. Before defining our design space for Internet connectivity in MANETs we review the related work in the context of the problem diagnosis.

# 3 Related Work

We classify related work in two main categories: 1) *Internet drafts* that describe a system framework or protocol. Some drafts that we have reviewed are outdated and are no longer easily found. They are therefore not addressed here. 2) Publications presenting an *evaluation of a system*, either a new or one from category 1. These almost exclusively focus on evaluating the overhead of different Mobile IP agent or gateway discovery approaches in simulation. Since this is not the focus of this paper, we leave most of them out.

Both categories have in common that the designs generally have not been implemented, except sometimes in simulation. Code is almost never available. Therefore they are hard to evaluate and the details of each system hard to grasp. We now review the categories in order.

## 3.1 Internet drafts

Belding-Royer et al. propose Globalv4 [5], which integrates Mobile IP with the AODV routing protocol. Globalv4 assumes flat addressing and hence a destination's location is determined by a local flood. If no route reply is received from the local search for a destination it is assumed that the destination is on the Internet. Delay is an obvious concern for this approach. In addition to flooding the network for determining node locations, gateway discovery is performed either by flooding the network with an agent solicitation, or by having the gateway periodically flood the network with agent advertisements. We find the latter approach odd considering the reactive routing. Since Globalv4 is targeted towards integration with Mobile IP it implies that there might be multiple MIP agents in the network acting as gateways. Therefore, we find it surprising that the routing approach taken is hop-by-hop and hence there is no way to enforce that data packets are routed through the same gateway as a node is currently registered at. That is because the routing protocol only cares about shortest paths and if another gateway suddenly is closer, the routing might update to reflect that. This update might go unnoticed by the source node if it happens on an intermediate node on the path to the gateway. We expect that this mismatch might occur occasionally until the views of AODV and Mobile IP are the same again. We have found no available implementations of Globalv4.

One of the most established Internet connectivity proposals are Globalv6 by Wakikawa et al. [19]. It is an

Internet draft targeted towards IPv6 networks for both reactive and proactive routing protocols. In Globalv6, nodes use one link local (MANET) address and one globally routable address for communication with the Internet. Intuitively this would double the overhead in the MANET because of the nodes' dual identities. Gateways are discovered through solicitation or gateway advertisement floods. In addition to this nodes must also flood the network once more to determine node locations in case of reactive routing. Forwarding to a gateway is done using the extended default route concept that we later describe in section 4.3.2. This concept gives raise to a number of problems, such as cascading route requests [13], mismatching route state in nodes and more. Only few of these problems are addressed in the draft. The details of these problems are explained and discussed in section 4.3. Despite being one of the most established proposals for Internet connectivity, we have only been able to find references to one available implementation of Globalv6 [9]. It is for the ns-2 simulator and implements parts of an early draft.

Jelger et al. [11] propose a system that ensures prefix continuity for MANETs that connect to the Internet through one or more gateways. All gateways announce their prefixes into the ad hoc network. Nodes carefully select addresses within the prefix of, e.g., the closest gateway, creating disjoint stub networks that share the same prefix. These stub networks contain a routing tree such that nodes can restrict themselves to storing a single default route. Such a system works best with proactive routing protocols and low network mobility. It uses IPv6 and targets specific scenarios where prefix continuity is important (e.g., Hot-spot operators). The focus of Jelger's work is a complement to other Internet connectivity solutions and lies outside the scope of the work presented in this paper.

### 3.2 Evaluations of Systems

Sun et al. describe in [17] a system that looks similar to Globalv4 (note the author overlap). They examine the effect of varying the Mobile IP agent beaconing interval for different network sizes. They also study the performance in terms of average packet latency and AODV overhead. Similar solutions for integrating MIP with ad hoc networks can be found in [18, 20].

Jönsson et al. studies in [12] the integration of Mobile IP in MANETs. They describe a system called MIPMANET where Mobile IP is adapted to work with MANETs running the AODV routing protocol. Tunneling from ad hoc nodes to the foreign agent is proposed as a way to achieve default route like behavior. However, the main result presented is the effect of using unicast or broadcast transmissions for periodic agent advertisements. We believe that periodic agent advertisements are not suitable for ad hoc networks using reactive routing. Ratanchandani et al. suggest in [16] a similar solution to MIPMANET. However, they also study the efficiency of agent discovery and suggest a hybrid approach where the TTL of agent announcements is used to limit propagation to a n-hop neighborhood. Nodes further away need to send agent solicitation messages to discover agents. In simulation they experimentally derive an optimal TTL for this approach.

Gateway discovery in on-demand MANETs is studied in [7], where Engelstad et al. examines problems with gateway proxy route replies in the presence of Network Address Translation (NAT). They find that tunneling to gateways is one way to avoid race conditions from proxy route replies when there are multiple gateways. This is in line with our findings as well.

Engelstad, Tønnesen, Hafslund and Egeland [8] study Internet connectivity in multi-homed proactive ad hoc networks. They also suggest tunneling to gateways for proactive routing and in particular to achieve multi-homing. The routing protocol's global view of the ad hoc network makes it easier to support Internet connectivity.

## 4 Design Space for MANET Internet Connectivity

In this section we explore the design space of MANET Internet connectivity to get a better understanding of the choices and trade-offs that are available for system designers. Table 1 shows the system elements and respective design choices that make up the design space. What follows is a discussion of each of the elements, their design choices and their applicability for the scenario described in section 2.

### 4.1 Determining a Node's Location

The problem of determining a node's location can be handled in different ways. If the MANET nodes share a common prefix or if the nodes have a global network view (e.g., as in proactive routing) they can determine node location by looking at the prefix of the destination or by investigating the routing table. However, for our scenario there must be an efficient way to handle node location. One approach suggested by Jönsson in [12] and Wakikawa in [19], is to flood the MANET with a route request. The lack of replies can be used by the source node as an indication that the destination resides in the Internet. This approach has obvious efficiency issues and increases the delay of route establishment.

A more efficient approach than timing out on the route request is suggested by Broch et al. [6]. A gateway could, in response to a route request, send a proxy route reply to signal that it can route to the requested destination. This

| System Element | Design Choice |
|---|---|
| Node location | *Prefix, flood, gateway, global view* |
| Gateway discovery | *Advertisement, solicitation, hybrid, integrated* |
| Forwarding | *Direct*: *host route, default route,* — *Indirect*: *tunneling, source routing* |

Table 1: Design Space for MANET Internet connectivity.

is analogous to the functionality of proxy ARP, but over multiple hops. To use the proxy-signaling, the gateway must determine that the destination is not in the ad hoc network. This can be done in different ways, including flooding the network with a new route request, by keeping a list of currently known active nodes (*visitor list*) or by pinging the destination on the gateway's network interface attached to the Internet.

## 4.2 Gateway Discovery

Gateways provide a service to other nodes in a MANET in that they offer Internet connectivity. Therefore, gateway discovery share similarities with service discovery in general. Two approaches dominate; service solicitation and service advertisement, but there are also hybrid ones that are a combination of both. A service solicitation corresponds to the reactive routing approach where nodes request information on-demand when needed. The service advertisement approach in turn corresponds to the proactive approach, where the provider of the service, in this case the gateway, regularly advertise its services into the network. In the context of MANETs there are obvious trade-offs with these two approaches. It does not make sense for a reactive MANET to integrate the advertisement approach and in the same way it does not make sense for a proactive MANET to use the solicitation approach.

Since a gateway in a MANET is just another ad hoc node, the most efficient design would be to integrate the gateway resolution process with that of path resolution in the routing protocol. The proxy gateway approach described in the previous section is an example of such a design.

Another design consideration is gateway election when there are several to chose from. Election can take place either at the gateways or at end-nodes. In the proxy approach, gateways can selectively reply to route requests depending on specific policies. A network operator might not want to announce its gateway services to some nodes, or an overloaded gateway might stop replying when the load reaches a certain threshold. Using end-node election, it is possible for nodes to use heuristics to chose a gateway, for example, depending on spatial proximity, load, or even to chose several gateways for multi-homing or load balancing. However, the MANET must also support forwarding to multiple gateways.

## 4.3 Forwarding

The forwarding approach employed in a MANET plays a crucial role for the flexibility of an Internet connectivity design.

### 4.3.1 Direct Forwarding Strategies

The direct forwarding strategies always do "shortest path" forwarding and do not diverge from the default path. The implications of Internet connectivity in this context is that forwarding is performed towards a destination outside the domain of the MANET. Direct forwarding is typically done hop-by-hop over a transient forwarding state installed in nodes between a source and destination. In the design space of MANET Internet connectivity we find two main approaches to do forwarding to gateways:

**Host Routes** A host route is a distributed state installed in all nodes comprising a path and consists of a number of consecutive mappings between a destination and a next hop. There is one set of mappings for each destination and hence there is no aggregation. Since the state is distributed it is possible to do local changes to the state and hence the path without all nodes on the path being aware of it. This also leads to a problem in the context of multiple gateways. It is not possible for end-nodes to enforce which gateway a route goes through. We described how this affects Globalv4 in section 3.

**Default Routes** The common notion of a default route is that from a Local Area Network (LAN) where it is a routing table entry pointing to the first router in the Internet. It represents the default next hop to send packets to that do not match any other explicit entry in a host's routing table. In contrast to a host route entry, which matches only one destination, a default route provides *aggregation* as it can map a number of destinations. In a MANET, where there may be several gateways located multiple hops away, the default route concept is not as applicable as in a LAN. There are two main issues: 1) The default route can only express the next hop, hence it is not possible to associate a specific gateway with the default route (Figure 1). 2) With reactive routing the aggregation

provided by the default route is lost because the lack of a routing table entry for the destination cannot be taken as a sign that a packet should be sent on the default route. This can lead to *cascading route requests* as mentioned in section 3.

#### 4.3.2 Extending the Default Route Concept

There are some suggestions how to extend the default route concept for use in MANETs. Figure 1 shows an example routing table for a node three hops from a gateway with address `192.168.1.1`.

| Destination | Next Hop | Hop Cnt |
|---|---|---|
| 63.3.5.23 | 63.3.5.23 | 1 |
| 66.35.250.151 | default | – |
| default | 63.3.5.23 | 3 |

(a)

| Destination | Next Hop | Hop Cnt |
|---|---|---|
| 192.168.1.1 | 63.3.5.23 | 3 |
| 63.3.5.23 | 63.3.5.23 | 1 |
| 66.35.250.151 | default | – |
| default | 192.168.1.1 | 3 |

(b)

Figure 1: Two examples of routing tables using a default route.

In (a) the default route is used without modification and maps to the *next hop* (`63.3.5.23`). Note that there is no explicit entry for the gateway and hence there is no way for this source node to know which gateway the default route leads to. In section 2 we explained that this can lead to interrupted connections. In (b), suggested in the Globalv6 proposal [19], the default route maps to the *gateway* address which in turn is used to find out the corresponding next hop. Here it is possible to know which gateway the default route leads to. However, this mapping state must also be consistent at each hop to the gateway. Both (a) and (b) need an extra host route entry for the destination to avoid subsequent route discoveries with reactive routing protocols. This state is installed when the node receives a route reply.

Although frequently suggested in related work, we find from our analysis that a default route is a concept that maps badly to MANET routing. The main reasons for this are: 1) With reactive routing the network must still be flooded to determine whether a destination should be associated to the default route. 2) The extra host route state that is associated with the default route removes the benefit of aggregation and combined with 1) the whole point of a default route is lost. 3) The host to default route mapping state needs to be replicated on all nodes between source and gateway. If the default route changes to include new intermediate nodes, those nodes must also be updated with all the host route state associated with that default route. We refer to this problem as the *state replication* problem and show in section 6 that this can lead to erroneous routing. 4) The extra mappings in the routing table adds extra overhead to the lookup process.

#### 4.3.3 Indirect Forwarding Strategies

An indirect forwarding strategy is one that allows non-shortest path forwarding of packets. This is useful if a source node wants packets to traverse a specific point on the way to the destination. This point can be, e.g., a proxy, Mobile IP home agent or in the case of MANETs a specific gateway. The reason to specify a gateway to traverse is that the gateway might have state, e.g, in the case of being a NAT or Mobile IP agent, that the source node is dependent upon. Hence, if the route to the Internet switches to another gateway without the knowledge of the source node, the connections that node maintains to the Internet will break. The main way to achieve indirection is by using source routing or flow based routing (as in MPLS). Another way is to use encapsulation, i.e., tunneling. In this section we will focus on tunneling since it has the benefit of enabling hop-by-hop forwarding to achieve indirection by encapsulating the packet in an extra IP header that specifies the indirection point. Figure 2 shows tunneling between a source node and a gateway in a MANET. The packets for the Internet destination is decapsulated at the gateway.

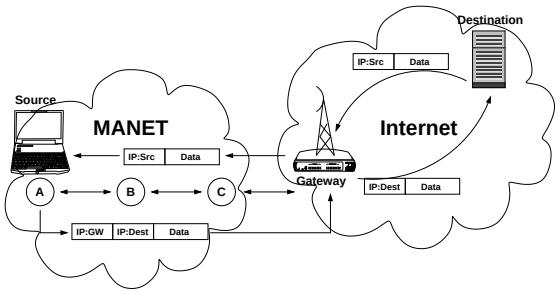


Figure 2: With tunneling the gateway becomes an indirection point. Packets for the Internet are encapsulated with a gateway's address and can be "locally" routed in the MANET between the source and gateway. Tunneling in the reverse direction is not necessary since the destination will then be a node in the MANET.

The obvious downside of a tunneling approach is the overhead and potential complexity of the encapsulation. However, tunneling also provides a number of benefits that we describe below:

- **Protocol transparency.** The tunneling concept is transparent with existing routing protocols. The minimum required modifications are extra routing table states in the source and gateway nodes which do not affect the protocol. There is no need for new state or routing modifications at intermediate nodes.
- **No cascading route requests.** Cascading effects are not a problem with tunneling because only the source node and the gateway need to know about the destination in the fixed Internet. Inside the ad hoc net-

work Internet packets are explicitly addressed for a gateway.

- **Route aggregation.** Tunneling achieves route aggregation at intermediate nodes since all Internet destinations are encapsulated by gateway addresses instead of one entry for each destination which is the case for host routes and default routes.

- **Stability.** Once a source node has configured a tunnel to a gateway, that tunnel will not be diverted to another gateway unless connectivity with the gateway is completely lost. In that case the source node will be notified and can take proper actions. For example, to re-register at a new gateway in case the source node is using Mobile IP.

- **Multiple gateways.** Source nodes can maintain routes to multiple gateways for fault tolerance and load balancing. Tunneling allow Internet traffic bound for different gateways over a common intermediate hop, which is not the case for default route forwarding (see Figure 3). Redundant tunnels can be used as as backup routes if the connectivity to one gateway is lost. This principle will avoid route request floods for all its current Internet destinations at connectivity loss. Furthermore, tunnels to multiple gateways are useful when a node wants to do a soft hand-over between gateways.

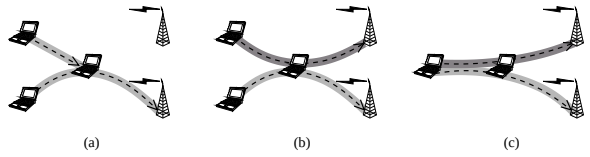


Figure 3: Multiple gateway support: (a) A default route points to only one gateway at once. (b) With tunneling two nodes can share an intermediate hop while still maintaining tunnels to different gateways, (c) or one node can have tunnels to two gateways at once.

- **Efficient forwarding.** In terms of routing table look-ups, tunneling is more efficient than the extended default route counterpart. A source node needs to perform two look-ups in the routing table. On intermediate nodes, only one regular look-up is needed.

There are other potential optimizations that can be achieved with tunneling, usually with the trade-off that it requires more tight integration with the routing protocol. For example, Intermediate nodes could be made gateway aware. An extra flag can be used to mark gateway routes in the routing tables. This can potentially avoid route requests if the source node directly can determine that its own packets should be tunneled to the Internet, e.g., by using a local prefix address.

### 4.4 Discussion

The analysis of different approaches in the design space of Internet connectivity provides a good starting point for system designers to better understand the trade-offs and problems. We believe this is important to construct more robust, efficient and clear designs that are easy to implement and in the end applicable to reality. There are also other things to consider in the design space that provide optimizations or flexibility in addition to robustness. One example is the ability to support multi-homing. Today it is not uncommon that mobile devices have more than one network interface to connect to the Internet through different access networks. For example, a laptop may have both a GPRS and a WiFi interface. These multi-homed devices could route packets over both interfaces at the same time to achieve smooth hand-over or load balancing. Or, in multi-homed sites there might be only one network interface, but multiple gateways in the same network. For similar reasons, connections to multiple gateways might be beneficial.

## 5 A Design for Internet Connectivity in MANETs

In this section we describe our design of Internet connectivity that builds on the design space analysis. We use AODV as the reference routing protocol to base our design on, because it matches the targeted scenario presented in section 2. We chose tunneling for our design because of the benefits described in section 4.3. For the purpose of the analysis of our design we also integrated default route forwarding as a reference in the comparison in section 6.

### 5.1 Implementation Details

We chose to implement the same type of gateway discovery and route setup mechanisms for both default routes and tunneling, so that a comparison would be as fair as possible. We adopted a proxy RREP solution (as suggested by Broch et al [6]) where the route discovery procedure of AODV is slightly extended to unify gateway discovery and route setup. This modification integrates well with AODV's reactiveness and is backwards compatible. A node initiates a route discovery by flooding the network with RREQs as it would normally do when it does not have a route to a destination. A gateway that receives this RREQ determines address locality (i.e., whether the destination is an Internet host or an ad hoc node) and will send a RREP to the ad hoc source node if the destination is an Internet host. The address locality check at the gateway is implemented through a prefix check or using a visitor list. The gateway's proxy RREP carries an extra AODV extension with the IP address of the requested Internet host

for which an ad hoc network node issued a RREQ. The RREP itself looks like a response to a RREQ for the gateway. This extended RREP is used to configure the default route or tunnel state at the same time as the gateway route is configured.

In the source node's routing table, gateways are marked with a `G` flag. Although not used for any purpose at this point, this flag could be used to indicate backup tunnels for faster hand-off. The RREP extension received from the gateway is used to configure an Internet host entry on the source node only. This entry points to the gateway, is marked with an `I` flag and has a limited life time. It maps fixed Internet addresses to the appropriate gateway addresses and represent tunnel entries. Our tunneling approach has been integrated into the AODV-UU [1] implementation. This code runs in both in simulation (ns-2) and on Linux systems. For the Linux version we use Minimal IP encapsulation (RFC 2004), which translates to an overhead of 8 bytes for each data packet sent through a gateway. We chose to implement default routes according to the Globalv6 draft [19]. Routing tables look like the one in Figure 1 (b). Host route entries on intermediate nodes are necessary to avoid cascading route look-ups. As an option we have also implemented a feature to sometimes drop conflicting RREPs that want to update a route to a new gateway. We show in the evaluation that this improves the performance of default routes, but does not solve all the problems.

Some optimizations still remain. At this point the tunneling implementation does not support intermediate node reply for Internet destinations. Another optimization would be to enable the use of backup tunnels. An Internet destination marked with an `I` flag, could easily be re-pointed to the next active tunnel that is marked with a `G` flag, without the need for a new RREQ flood.

# 6 Evaluation

In this section we compare two Internet connectivity designs with a focus on forwarding strategies. We present simulation results that compare the performance of default route and tunnel forwarding using constant bit rate (CBR) UDP traffic and (FTP) TCP traffic. We show that tunnel forwarding constantly performs better than the default route route counterpart. The simulations provide the main result of this paper. We then continue to present results from our experimental testbed. The purpose is to provide a proof-of-concept of how the tunneling approach can function together with AODV and Mobile IP in a real system.

## 6.1 Simulations

We use *ns-2* version 2.26 and the ns-2 AODV-UU implementation of AODV. We have gateway forwarding for both default routes and tunneling. We chose to evaluate gateway forwarding in network scenarios where we scale the number of nodes from 10 to 20 mobile nodes, incrementing by two at a time. Two gateways are used in the simulations. We keep node density fixed at $2 \times 10^{-5}$ nodes per $m^2$. Thus, the area size (with an x:y ratio of 1:2) grows with increased number of nodes. We found this density to be a good balance between network size and number of nodes. We have one mobile node per 223 m square (the two gateway nodes excluded) and with the default ns-2 transmission radius of 250m (using the "TwoRayGround" model), we have reasonable coverage as indicated by the performance figures. With a fixed density, routes on average are likely to be longer as the number of nodes and area size increase. This allows us to evaluate forwarding behavior with increasing path length. The ad hoc nodes move in the simulated area according to the *random waypoint* model with a max speed of 20 $m/s$. We randomly generated 50 movement pattern files for each network size (i.e., number of nodes and area size). One movement run lasts for 200 seconds. These 50 patterns were used for all experiments with default routes and tunneling to ensure that the results were comparable. Performance averages were taken over all 50 runs for each network size and forwarding strategy.

### 6.1.1 CBR Performance

In our first experiment we examine CBR performance because the traffic is predictable and there is no feedback loop, i.e., no acknowledgments like in TCP. This also means that it does not matter to which gateway the traffic is forwarded, since there is no return traffic. We would therefore not expect dramatic differences between strategies. Two CBR sources, sending 512 byte packets at a rate of 10 per second, is randomly selected among the ad hoc nodes. They will start to communicate with a host on the fixed network by randomly choosing one out of two possible. The CBR sources start 5 seconds into the simulation and continue until the end at 200 seconds.

In Figure 4 we see a comparison of aggregated delivery ratios (data packets received divided by data packets transmitted) for tunnel forwarding and default route forwarding. Although the variance for each point in the diagram is quite high, the differences between curves are significant. The variance is caused by the randomness in movement patterns. As can be seen from the left figure the delivery ratio decreases with the number of nodes. This is expected since the number of hops to the gateway will also increase with the size. Hence will the probability of connectivity loss also increase with hop length and consequently the control traffic will increase to handle the losses. This overall pattern occur in all our measurements.

We note that tunnel forwarding consistently achieves better delivery ratio than the default route approach. Our

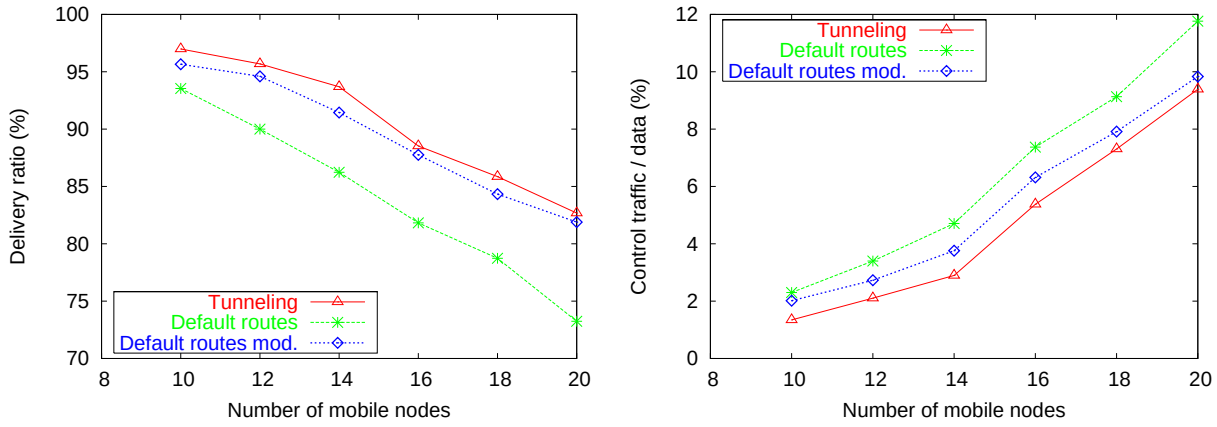


Figure 4: CBR Delivery ratio and normalized control traffic overhead using 2 CBR sources and increasing path lengths.

hypotheses is that the default route solution occasionally suffers from the state replication problem (incorrectly replicated or missing state on the nodes along a default route). If there is missing state, the AODV protocol will send a route error message to upstream nodes to invalidate their default routes and force them to be rediscovered. This would explain the larger amount of control message overhead of default route forwarding compared to tunneling in the right graph of Figure 4. It is likely that missing state is more frequent when we have more nodes and a larger simulated area, thus on the average longer routes. The delivery ratio in Figure 4 supports this view, since the gap between tunnel forwarding and default route forwarding increases with more nodes. To verify our hypotheses we changed the AODV implementation so that it always forwards packets on a configured default route. This will work because we only have traffic for the two fixed hosts and it provides a reference for how default routes should work without state replication problems. However, this "hack" is not useable in real life. To mitigate the problem of not being able to enforce the usage of a particular gateway, the modification also drops route replies that try to reconfigure an existing default route to point to another gateway. The simulations with these modifications are called "default route mod." in the figures and show that the modifications bring the CBR delivery ratio much closer to tunnel forwarding as expected.

### 6.1.2 TCP Performance

For TCP it is important that the return traffic (i.e., the acknowledgments) from the fixed network is sent through the same gateway as the forward traffic. Otherwise TCP acknowledgments might get lost in case they are sent to a gateway which does not have connectivity with the source node. To support this we modified ns-2's Mobile IP to work with the AODV implementation. Mobile IP's agent discovery was removed and replaced with AODV's RREQ mechanism just to simplify the set-up. For other parts Mobile IP works as specified. Whenever a mobile ad hoc node selects or switches gateway it will register with the agent at that gateway so that return traffic is delivered there.

For this experiment, our scenario's two gateways are assigned as Home Agent (HA) and Foreign Agent (FA), respectively. The scenario configuration otherwise remains the same. We created two FTP sources. The aggregated throughput of these two is limited by the gateway capacity.

In Figure 5, we see that the expected drop in performance with the number of nodes, caused by the increased probability of losing connections. When comparing strategies we see that tunnel forwarding constantly achieves a higher TCP throughput than default route forwarding. The lower throughput for default routes is likely caused by a change in the default route → gateway mapping somewhere on the path to a gateway without the source node being notified. Consequently, the source node never registers at the new gateway and the acknowledgments might be lost, resulting in a TCP timeout.

The TCP goodput (ratio of TCP packets successfully delivered to the total number of TCP packets transmitted) in the right figure and the control traffic overhead in Figure 6 supports this view. Surprisingly, the goodput of tunnel forwarding is slightly lower than that of default route forwarding. Although the difference is small enough to fall within the error margin, one explanation is that with less timeouts for tunneling it will send more packets than default route and thus also retransmit more packets. This would reduce the goodput of tunneling while it still has a higher throughput than default route. At the same time, default route forwarding is not retransmitting that much, indicating that the decreased throughput is caused by timeouts. Control traffic is also likely to increase, since with tunnel forwarding, AODV spends more time delivering packets than idle in timeouts. In combi-

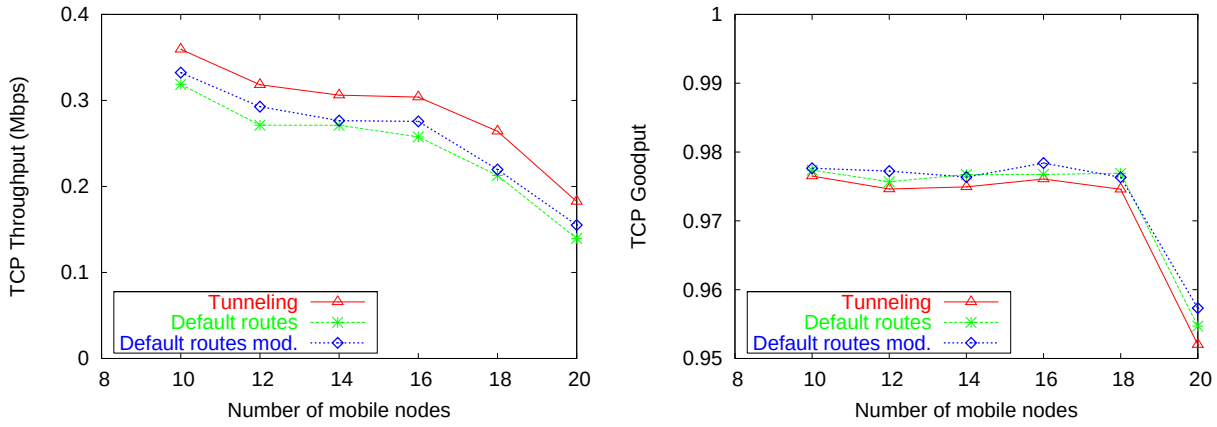


Figure 5: TCP throughput and goodput using 2 TCP sources and increasing path lengths.

nation this will give less goodput for tunneling. Interesting is that the modified default route forwarding does not show a similarly strong improvement in this experiment as in the CBR case. This is in line with our assumptions that default route forwarding does not work well with TCP in multiple gateway scenarios. We will explore this issue further in the next section.

### 6.1.3 Maintaining Consistent Gateway Connectivity

In section 4.3 we described the inability to "track" gateways with both host routes and default routes. With Mobile IP, return traffic should be sent to the gateway at which the ad hoc source node is currently registered. If this is a different gateway from the one that forward traffic is sent over, the TCP acknowledgments might be stuck there if there is no alternative path. This could explain why TCP with default route forwarding seems to spend more time idle in timeouts.

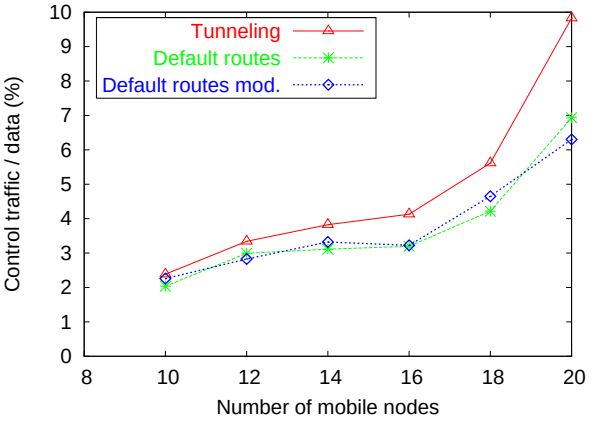


Figure 6: Normalized control traffic overhead in TCP scenario.

We wanted to verify with an experiment that the low throughput is caused by timeouts and whether dropping conflicting route replies solves the problem. We constructed a scenario where the routing protocol is subjected to frequent gateway changes. A mobile node (MN) communicating with a host on the fixed network moves back and forth, forcing change in connectivity between two gateways (a HA and a FA). Connectivity with the gateways are always possible through intermediate nodes so that the hop count to the closest gateway is always three. This scenario was created to mimic the experimental setup we have used for our real world experiments presented in section 6.2. Figure 8 illustrates this scenario.

MN will initiate an FTP file transfer to the fixed host at the start of the simulation and lasts 200 seconds. During this time MN will move back and forth twice, with equal connectivity to both gateways at times 25, 75, 125 and 175 seconds. Thus a gateway change should be triggered around those times. The other nodes remain stationary and will forward traffic to and from the gateways.

Figure 7 shows a TCP sequence number trace of a simulation run. In this scenario we expected to see time gaps in between sequence numbers corresponding to hand-over points. Tunnel forwarding reaches the highest sequence number at 200 seconds. There are expected gaps for tunneling during gateway changes, but they are not so visible due to random effects on TCP. The unmodified default route forwarding on the other hand has two long periods where there are no packets sent at all and TCP timeouts. The first timeout corresponds well to the time of the first gateway change and the other with the third gateway change. It seems as if traffic is only forwarded over one of the gateways (the home agent).

We wanted to find out the exact cause of this behavior and therefore examined our log files. We found the following explanation: With unmodified default routes, route replies from both gateways at hand-over installs conflicting state, updating the gateway mapping on intermediate nodes while not propagating correctly to MN.

MN believes the Internet host can be reached through the HA, when in reality the packets are forwarded through the FA. Since the forwarding to the fixed host still works, MN will keep and continue refreshing its default route pointing to the HA. MN incorrectly concludes that it does not have to register at the FA, causing the acknowledgments to be lost at HA. This will continue until MN loses the connectivity to the FA and regains connectivity with the HA. Thus, TCP will go into a timeout.

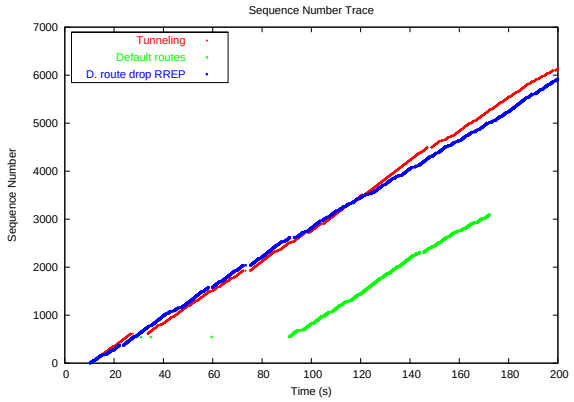


Figure 7: Sequence number trace showing how default routes have problems with multiple gateways.

Modified default routes drop route replies conflicting with a configured default route. In the resulting sequence number trace we see that this solves the problem as expected. In this isolated case the route reply is the culprit. However, dropping these conflicting route replies seems to have little effect in the general case as we experienced from the CBR and TCP simulations. Thus we conclude that the state replication problem has a bigger impact on the performance than the gateway tracking problem.

## 6.2 Experimental Results

In this section we illustrate the functionality of our real world implementation of Internet connectivity using AODV and Mobile IP. We have already investigated our designs ability to perform well and correctly in simulation. The purpose here is therefore to show that our system works in the real world as well and can interface with Mobile IP to provide uninterrupted TCP connections to the Internet while changing gateway.

We have implemented an experimental testbed using our design with tunnels toghether with Mobile IP. Our mobile host (MH) is a IBM T30 2.0 GHz laptop running Linux while the rest of the ad hoc network nodes (CH) are IBM T31 laptops also running Linux. The foreign agents (FAs) are LinkSys WRT54G routers running the OpenWRT Linux distribution with kernel 2.4.30. All ad hoc nodes run the AODV-UU implementation with our Internet connectivity. The Home agent (HA) is a 2.4 GHz Pentium 4 desktop Linux machine. To provide continuous reachability for the MH, Mobile IP is used to redirect the MHs return traffic whenever it changes its location, i.e., registers with a new FA. The HUT Dynamics Mobile IP implementation [2] was chosen for this purpose. We had to make slight modifications to the Mobile IP implementation to interoperate better with the AODV ad hoc routing.

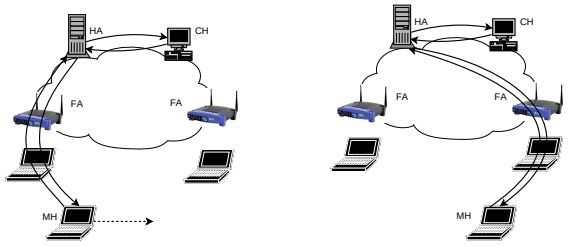


Figure 8: Example scenario in our experimental testbed setup using LinkSys routers. The mobile host (MH) has gateway connectivity over the ad hoc network. Communication with the correspondent host (CH) is established through a bidirectional tunnel between a foreign agent (FA) and the home agent (HA). When a new gateway is discovered, the MH re-registers with the FA at that gateway and the bidirectional tunnel is re-pointed to the new FA.

The integration of AODV and Mobile IP was implemented as follows. We disabled all proactive agent advertisements in the FAs. Whenever the MH floods the network with a AODV route request, the FAs will answer with an extended route reply (described in the previous section) indicating that this Internet destination can be reached through the FA. Immediately after this reply, the AODV daemon on the FA triggers the MIP daemon to send an agent advertisement to the MH on the newly established route. The AODV daemon on the MH will force its Mobile IP daemon to select the FA that matches the gateway selected by AODV. When the MH receives the agent advertisement it will send a registration message to the selected FA that will configure a bidirectional tunnel between the FA and the HA. The experimental testbed configuration is illustrated in figure 8. We ran a number of experiments using both bidirectional Ping traffic and TCP file transfers with varying number of hops to the gateway. Due to space limitations we present only one test using TCP. The experiment lasts for 60 seconds and during that time a mobile node moves from one gateway acting as a FA to another FA gateway while transferring data using TCP to a correspondent host on the Internet, via its home agent.

Figure 9 shows the sequence number trace for an example run. The time of mobility is identified by the periods of jaggedness in the trace. The gateway change occurs around time 35 s, as can be seen by the slight interruption

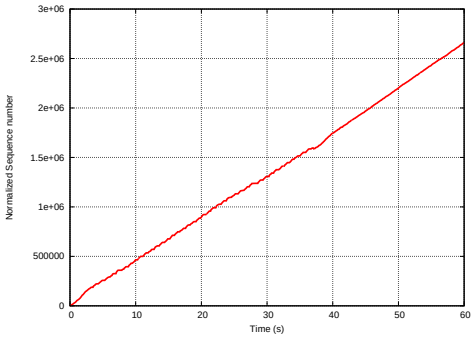


Figure 9: Sequence number trace from experimental test showing the TCP progress while switching gateways.

in TCP's progress.

# 7 Conclusions

We have diagnosed the problems of providing Internet connectivity in MANETs and from that we have defined a design space. We argue that this is important because Internet connectivity is a very compelling service that has the potential to increase the benefits of MANETs and hence their deployment in real life. A design space will help other potential designers to better understand the trade-offs and problems of the different design choices. It also provides a framework for evaluating existing Internet connectivity designs.

One important and distinguishing point in our diagnosis is that it builds on a scenario that is as challenging as possible. The lack of targeted scenarios for many designs make them hard to understand and evaluate. A clear and challenging scenario enforces robustness in the design choices and resulting solution also works for less challenging scenarios. In that sense we have a based our reasoning around a "worst case". We have surveyed the existing solutions to Internet connectivity and have found that most of them are not robust enough, not even for less stringent scenarios. Furthermore, many of the solutions are only proposals and have not been implemented and evaluated, in particular side-by-side, neither in simulation nor in reality.

We have compared two solutions in the design space, one using default routes and one using tunneling. From our analysis we found that the commonly proposed default route solutions to Internet connectivity lack the ability to, among other things, express *indirection*. On the other hand, an indirect forwarding approach using tunneling, does not suffer from the default route's inherent problems and is also more flexible in terms of multi-homing support and is also more efficient. Our conclusion is that tunneling or other approaches to express indirection are required to build efficient and robust Internet connectivity designs for MANETs. From our analysis and the conclusions from the evaluation we believe that our complete design for Internet connectivity is simpler and more efficient than other approaches that have not been implemented or properly evaluated. We have illustrated through experiments how our system can operate robustly together with Mobile IP in a multiple gateway environment and how TCP sessions can be maintained while switching between different gateways.

# References

[1] The AODV-UU implementation. `http://www.docs.uu.se/scanet/aodv`.

[2] Dynamics Mobile IP. `http://dynamics.sourceforge.net`.

[3] RoboCupRescue. `http://www.rescuesystem.org/robocuprescue/`.

[4] Rotondus: Durable mobile robots for outdoor surveillance. `http://www.rotundus.se`.

[5] E. Belding-Royer, Y. Sun, and C. Perkins. Global connectivity for IPv4 mobile ad hoc networks, November 2001. IETF Internet Draft, draft-royer-manet-globalv4-00.txt, (work in progress).

[6] J. Broch, D. A. Maltz, and D. B. Johnson. Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks. In *Proceedings of the Workshop on Mobile Computing*. IEEE, 1999.

[7] P. Engelstad and G. Egeland. NAT-based Internet connectivity for on-demand ad hoc networks. In *WONS 2004*, pages 344–358, 2004.

[8] P. Engelstad, A. Tønnesen, A. Hafslund, and G. Egeland. Internet connectivity for multi-homed proactive ad hoc networks. In *The IEEE International Conference on Communications (ICC) 2004*, June 2004.

[9] A. Hamidian. A study of internet connectivity for mobile ad hoc networks in ns 2. Master's thesis, Lund Institute of Technology, January 2003.

[10] I. S. I. S. interest group. nterplanetary internet project. `http://www.ipnsig.org`.

[11] C. Jelger, T. Noel, and A. Frey. Gateway and address autoconfiguration for ipv6 adhoc networks, October 2003. IETF Internet Draft, draft-jelger-manet-gateway-autoconf-v6-01.txt.

[12] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. Maguire Jr. MIPMANET - Mobile IP for Mobile Ad hoc Networks. In *1st ACM international symposium on Mobile ad hoc networking and computing (Mobihoc'00)*, 2000.

[13] A. Nilsson, C. E. Perkins, A. J. Touminen, R. Wakikawa, and J. T. Malinen. AODV and IPv6 Internet access for ad hoc networks. *Mobile Computer and Communications Review*, 6(3):102–103.

[14] C. Perkins. IP mobility support for IPv4, January 2002. IETF Internet RFC 3220.

[15] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, July 2003. IETF Internet RFC 3561.

[16] P. Ratanchandani and R. Kravets. A hybrid approach to internet connectivity for mobile ad hoc networks. In *IEEE WCNC*, 2003.

[17] Y. Sun, E. M. Belding-Royer, and C. E. Perkins. Internet connectivity for ad hoc mobile networks. *International Journal of Wireless Information Networks*, 9(2):75–88, April 2002.

[18] Y.-C. Tseng, C.-C. Shen, and W.-T. Chen. Integrating mobile IP with ad hoc networks. *Computer*, (5):48–55, 2003.

[19] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. Tuominen. Global connectivity for IPv6 mobile ad hoc networks, (work in progress), July 2005. IETF Internet Draft, draft-wakikawa-manet-globalv6-04.txt.

[20] C. Åhlund and A. Zaslavsky. Software solutions to Internet connectivity in mobile ad hoc networks. In *4th International Conference on Product Focused Software Process Improvement (PROFES)*, 2002.